Mandatory Handin 4 - Distributed Mutual Exclusion

Nicolaj Heinrich Pedersen (nihp), Mikkel Bistrup Andersen (mbia)

November 2022

`https://github.com/mbia-ITU/DISYS-HandIn-4`
IT-University of Copenhagen

Contents

1 Assignment 4 - Peer to Peer

1.1 Discussion of Algorithm

We have chosen to use Ricart & Agrawala's algorithm for prioritizing access to the critical section. This algorithm is the most comprehensive since it check with every single peer if it has priority to the critical section and if not it yields and gives another node access first. This way we ensure that two nodes never has access to the critical section at the same time. This can be seen in the *logs* section of our report.

1.2 Copy of readme

How to use program
1: open up a terminal.
2: move into folder and type "go run peer2peer.go" follow by a non-negative integer between 0 and 2.
3: open another terminal and follow step 2, but with a different integer for each time.
4: press enter to ping all connected peers.
5: close terminals when done.

1.3 logs

```
PS C:\Users\Nicol\Documents\A Folder for Uni\GOP2P\DISYS-HandIn-4> go run peer2peer.go 0
Trying to dial: 5001
successfully dialed: 5001
Trying to dial: 5002
successfully dialed: 5002

Requesting access to critical section from other cliets...
Got reply from id 5001: you're free to go
Got reply from id 5002: I was first! (id prio)
Resending request...
Got reply from id 5001: you're free to go
Got reply from id 5002: you're free to go
5000 just accessed the critical section at timestamp 8
```

```
PS C:\Users\Nicol\Documents\A Folder for Uni\GOP2P\DISYS-HandIn-4> go run peer2peer.go 1
Trying to dial: 5000
successfully dialed: 5000
Trying to dial: 5002
successfully dialed: 5002

Requesting access to critical section from other cliets...
Got reply from id 5000: I was first! (timestamp prio)
Resending request...
Got reply from id 5000: you're free to go
Got reply from id 5002: you're free to go
5001  ust accessed the critical section at timestamp 6
```

```
PS C:\Users\Nicol\Documents\A Folder for Uni\GOP2P\DISYS-HandIn-4> go run peer2peer.go 2
Trying to dial: 5000
successfully dialed: 5000
Trying to dial: 5001
successfully dialed: 5001

Requesting access to critical section from other cliets...
Got reply from id 5000: you're free to go
Got reply from id 5001: you're free to go
5002 just accessed the critical section at timestamp 4
```

As seen in the image above, three clients connect to each other, as soon as

they were connected, a request to access the critical section was sent from each of them, client 5002 wins the first priority to enter from all of them, since it has id prio (id prio was reversed when the picture was taken, this has been fixed in the newest build), next to gain access is client 5001, this is mostly due to how delay is simulated via time.sleep, it manages to finish before client 5000 sends its re request fully, though it would have normally won anyways with timestamp prio to client 5000, and lastly client 5000 finishes since no other clients remain trying to access the critical section.